Secondary 1

Programming and AI

Eng. Osama Youssef

WhatsApp: 01551212333

Chapter 18: HTML

Hyper Text Markup Language
Used to build the structure of a website

Eng. Osama Youssef
01551212333

Tags

START TAG — CONTENT — END TAG

```
1  <h1>CONTENT</h1>
   <p>Normal Paragraph text</p>

2  <p>A simple sentence.</p>
```

```
Bread
Ham
Cheese
```

```
Bread<br>
Ham <br>
Cheese
```

Visualized Outputs

```
Bread
Ham
Cheese
```

```
Bread Ham Cheese
```

Heading Levels

HTML Essentials: Lists

THE KEY RULE

- use <ul> + for bulleted lists
- use <ol> + for numbered lists

Unordered List

- Item 1
- Item 2
- Item 3

Ordered List

1. Item 1
2. Item 2
3. Item 3

Cooking Wishlist

- Tomato
- Onion
- Mushroom
- Garlic

School Subjects

1. Science
2. Math
3. Music
4. History
5. Physical I

NOTES

HTML ESSENTIALS: LISTS & IMAGES

Eng. Osama Youssef
01551212333

<ul

Unordered List

Ordered List

1. 2. 3.

THE KEY RULE

1. use <ul> + for bulleted lists,
2. use <ol> + for numbered lists

IMAGE TAG ANATOMY

src="image.jpg" — alt="Description"

UNORDERED LIST EXAMPLE

```
<ul>
  <li>Tomato</li>
  <li>Onion</li>
  <li>Mushroom</li>
  <li>Garlic</li>
</ul>
```

BROWSER OUTPUT: FOODS

- Tomato
- Onion
- Mushroom
- Garlic

ORDERED LIST EXAMPLE

```
<ol>
  <li>Science</li>
  <li>Math</li>
  <li>Music</li>
  <li>History</li>
  <li>Physical Education</li>
</ol>
```

BROWSER OUTPUT: SUBJECTS

1.science
2.Math
3.Mussic
4.History
5.Physical Education

COMPLEX INTEGRATION

```
<h1>Cooking Wishlist</h1>
<ul>
  <li>Curry rice <img
      src="curry.jpg"
      alt="Curry rice" </li>
  <li>Omelette rice <img
      src="omelette.jpg"
      alt="Omelette rice" </li>

  <li>Pasta</li> <br>
<img src="pasta.jpg"
      alt= Tomato Sauce pasta">
  </li>
```

Successful Render: Cooking

Cooking Wishlist

- Curry rice
- Omelette rice

BROKEN RENDER: Pasta

Pasta

Tomato sauce pasta

NOTES & COMMENTS

The <img> tag, like <br>, does not need an end tag.

src and alt are examples of attributes that add extra information to the tag.

To write a comment:

< !-- comment -->

NOTES

CHAPTER 19: CSS (CASCADING STYLE SHEETS)

Eng. Osama Youssef
01551212333

Core Syntax Module

```
SELECTOR { PROPERTY: VALUE; }
```

Basic CSS Styling

```
h1 {
   color: blue;
}
p {
   color: red;
}
```

```
h1 {
   color: red;
   background-color: yellow;
p {
   color: blue;
   background-color: orange;
}
```

Typography & Classes

P

font-weight values:
bold,normal, and 100-900

font-size using "px"

```
p { font-weight: bold }
p { font-size: 50px; }
```

```
<p class="apple">Apple</p>
<p class="watermelon">Watermelon</p>

.apple{ color: red; }
.watermelon{ color: green; }
```

You can write a comment using /* COMMENT */.

NOTES

Conditions (IF Statement)

Eng. Osama Youssef
01551212333

Remember:

If the conditions should be **true**, then you need to proceed with the specified action.

Logical Operators

Symbol	Name	meaning	Example
&&	AND	Both conditions should be true	`if (age > 18 && score >= 50)`
||	OR	Only one condition should be true	`if (age >= 18 || age < 18)`

Comparison vs. Assignment

1. We use **"="**: to assign value to a variable.
2. We use **"=="**: to compare two values.

Case 1: 1 Condition - 1 Action

```
if (condition) {
   action
}
```

Case 2: 1 Condition - 2 Actions

```
if (condition) {
   action 1
} else {
   action 2
}
```

Case 3: Multiple Conditions

```
if (condition) {
   action 1
} else if {
   Action 2
} else if {
   Action 3
.
.
} else {
   Action 4
}
```

JavaScript — Run Code

```
let science = 100;
if (science == 100) {
  console.log("Excellent");
}
```

Console

```
Excellent
```

JavaScript — Run Code

```
let math = 80;
if (math >= 90) {
  console.log("Excellent");
} else {
  console.log("You need to study hard");
}
```

Console

```
You need to study hard
```

NOTES

SCOPE OF VARIABLES

Eng. Osama Youssef
01551212333

Scope: where a variable can be used

BLOCK SCOPE { }

Variables declared inside {} of an `if` , `else`, or other block cannot be used outside those {}.

```
let contest = 1;
if (contest == 1) {
  let reward = 30000;
  console.log(reward + " codycoins GET");
}
```

Output: 30000 codycoins GET

```
let contest = 1;
if (contest == 1) {
  let reward = 30000;
}
console.log(reward + " codycoins GET");
```

Output: Error (reward is not defined)

```
let contest = 5;
if (contest == 1) {
  let reward = 30000;
  console.log(reward + " codycoins GEI");
} else {
  console.log(reward + " codycoins on hold");
}
```

Output: Error

GLOBAL SCOPE

Variables declared outside of any block scope {} can be used both outside and inside blocks.

```
let science = 100;
let bonus = 5000;
```

```
let science = 100;
let bonus = 5000;
if (science == 100) {
  console.log(bonus + " codycoins GET");
}
```

Output: 5000 codycoins GET

NOTES

NESTED IF

OVERVIEW & RULES

Rule 1:
You can put an if statement **INSIDE** another if statement.

Rule 2:
The inner if runs **ONLY** when the outer if condition is **TRUE.**

Rule 3:
Place the inner if and its outputs **INSIDE** the { } of the outer if.

SCENARIO 1

JavaScript

```javascript
let score = 90;
let rank = 1;
if (score >= 80) {
  console.log("Test Passed");
  if (rank == 1) {
    console.log("Trophy Acquired");
  }
}
```

OUTPUT

```
Test Passed
Trophy Acquired
```

SCENARIO 2

JavaScript

```javascript
let score = 90;
let rank = 3;
if (score >= 80) {
  console.log("Test Passed");
  if (rank == 1) {
    console.log("Trophy Acquired")
  }
}
```

OUTPUT

```
Test Passed
```

SCENARIO 3

JavaScript

```javascript
let score = 75;
let rank = 1;
if (score >= 80) {
  console.log("Test Passed");
  if (rank == 1) {
    console.log("Trophy Acquired");
  }
}
```

OUTPUT

```
NO Output
```

Eng. Osama Youssef
01551212333

NOTES

Nested Conditionals

Quick Tip: Pay close attention to indentation and closing braces { }.

What are Nested Conditionals?

1. An 'if' statement placed 'inside' another 'if' statement.
2. Creates more detailed and layered decision logic.

Case 1: Primary and Secondary Checks

Run Code

```
let weather = "sunny";
let shop = "OPEN";

if (weather == "sunny") {
  if (shop == "OPEN") {
    console.log("Let's buy bread and go for a picni
  }
}
```

Console

Let's buy bread and go for a picnic

Case 2: Handling Specific Alternatives

Run Code

```
let weather = "sunny";
let shop = "CLOSED";

if (weather == "sunny") {
  if (shop == "OPEN") {
    console.log("Let's buy bread and go for a picnic");
  } else {
    console.log("Let's buy apple and go for a picnic");
  }
} else {
  console.log("Lunch at home");
}
```

Console

Let's buy apple and go for a picnic

Case 3: Multi-Layer Decision

Run Code

```
let weather = "sunny";
let group = "Group A";

if (group == "Group A") {
  if (group == "Destination is the zoo") {
    console.log("Destination is the museum");
  }
} else {
  console.log("Field trip canceled");
}
```

Console

Destination is the zoo

Case 4: Multi-Path Complexity

```
let weather = "sunny";
let group = "Group C";
if (weather == "sunny") {
  if (group == "Group A") {
    console.log("Destination is the zoo");
  } else if (group == "Group B") {
    console.log("Destination is the museum");
  } else {
    console.log("Destination is the art museum");
  }
} else {
  console.log("Field trip canceled");
```

Console

Destination is the art museum

Eng. Osama Youssef
01551212333

NOTES

Discover the mistake in case 4

Arrays

Values in Arrays

Values in arrays start from index 0.
Ex: To get one value: tempList[2]

```
1 let tempList = [36.2, 36.5, 36.3, 36.8, 36.4];
2 console.log(tempList[2]);
```

Output: 36.3

Combining Loops & Arrays

```
1 let tempList = [36.2, 36.5, 36.3, 36.8, 36.4, 36.9, 36.6];
2 let sum = 0;
3 for (let i = 0; i < 7; i++) {
4   sum = sum + tempList[i];
5 }
```

Output: 255.70

Arrays with Loops & If

```
1 let tempList = [36.0, 36.5, ...];
2 for (let i = 0; i < tempList.length; i++) {
3   if (tempList[i] >= 37.0) {
4     console.log("Be careful!");
5   }
6 }
```

JavaScript

```
1 let tempList = [36.2, 36.5, ...];
2 let sum = 0;
3 for (let i = 0; i < 7; i++) {
4   sum = sum + tempList[i];
5 }
2 console.log(sum / 7);
```

(Average)

JavaScript

```
for (let i = 0; i < tempList.length; i++) {
  ...
}
console.log(sum / tempList.length);
```

(Average)

Eng. Osama Youssef
01551212333

NOTES

Looping

Eng. Osama Youssef
01551212333

General Loop Syntax

```
for (start; condition; update) {

}
```

Case 1: Simple Loop Execution

JavaScript — Run Code

```
for (let i  = 0; i < 3; i++ ) {
  console.log("She sells seashells by the seashore.");
}
```

Console — Console

```
She sells seashells by the seashore.
She sells seashells by the seashore.
She sells seashells by the seashore.
```

Scope of for loop

Variables declared *inside* the parentheses ( ) of a `for` loop (like the counter `i`) cannot be accessed *outside* its curly braces { }.

Case 2: Accessible Counter

JavaScript — Run Code — Console

```
for (let i = 0; i < 3; i++) {
   console.log(i);
}
```

i exists here.

Output:
0
1
2

Case 3: Out of Scope Error

JavaScript — Run Code — Console

```
for (let i = 0; i < 3; i++) {
   console.log(i);
}
console.log(i);
```

i is not accessible here.

Output:
0
1
2

Uncaught ReferenceError: i is not defined

Case 4: Variables Declared Inside Loop

JavaScript — Run Code — Console

```
for (let i = 0; i < 3; i++) {
   let num = 100;
   console.log(num);
}
```

New "num" variable is created on each iteration.

Output:
100
100
100

NOTES

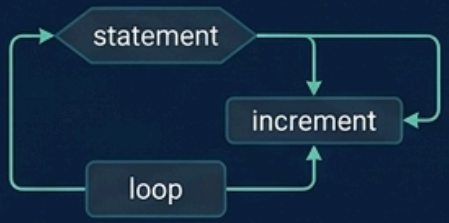
NESTED LOOPS & CONDITIONALS

Understanding for loops with if statements and mathematical operations.

Eng. Osama Youssef
01551212333

KEY SYNTAX & CONCEPTS

We use i++ to mean i = i + 1



CASE 1: JAVASCRIPT

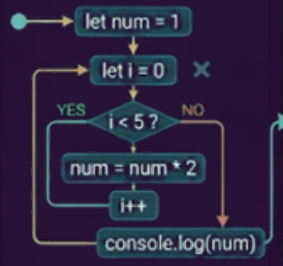
```
for (let i = 0; i < 5; i++) {
  console.log("sit-up");
  if (i == 2) {
    console.log("running");
  }
}
```

CONSOLE

```
sit-up
sit-up
sit-up
running
sit-up
sit-up
```

CASE 2: CALCULATE REPEATED MULTIPLICATION

```
let num = 1;
for (let i = 0; i < 5; i++) {
  num = num * 2;
}
console.log(num);
```



CONSOLE

```
32
```

CASE 3: MULTIPLY NUMBER OF i

```
let num = 1;
for (let i = 1; i <= 5; i++) {
  num = num * i;
}
console.log(num);
```

CONSOLE

```
120
```

NOTES

Eng. Osama Youssef
01551212333

📖 *To print all values in an array, use a variable i*

1. Static Loop:

JavaScript | Run Code | Console

```
let degreeList = [2, 1, 2, 1, 0, 1, 1];
for (let i = 0; i < 7; i++) {
  console.log(degreeList[i]);
}
```

Output: Run Code

```
2
1
2
1
0
1
1
```

Eng. Osama Youssef
01551212333

2. Dynamic Loop (using .length):

JavaScript | Run Code | Console

```
let degreeList = [2, 1, 2, 1, 0, 1, 1];
for (let i = 0; i < degreeList.length; i++) {
  console.log(degreeList[i]);
}
```

Output: Run Code

```
2
1
2
1
0
1
1
```

3. for...of Loop:

JavaScript | Run Code | Console

```
let degreeList = [8, 6, 11, 10, 9, 7, 12];
for (let degree of degreeList) {
  console.log(degree);
}
```

Run Code

Output:

```
8
6
11
10
9
7
12
```

NOTES

Functions in JavaScript

Eng. Osama Youssef
01551212333

1. What is a Function?

```
function _____() {
  ...
}
```

Functions (with arguments)

```
function divide6(total) {
  let per = total / 6;
  return per;
}

console.log(divide6(72));
```

Console/Output: 12

Receives 'total', executes division, returns 'per'.

Functions (without arguments) & Multi-parameter

```
function goodMorning() {
  let greetin
  return gree
}

console.log(g
```

Console/Output: Good morning!

osama said

```
function sum(
  let result = num1 + num2;
  return result;
}

console.log(sum(50, 100));
```

Console/Output: 150

Variable Scope

```
function split5(total) {
  let result = total / 5;
  return result;
}

console.log(result);
```

Console/Output: Error: result is not defined

Variable 'result' is only accessible within the split5 function.

NOTES